CS112 - Fall 2022

Lab22

Instructor: Paul Haskell

# INTRODUCTION

In this lab, you will work with recursion.

## Euclid's Algorithm

Suppose you have two positive integers, A and B. The greatest common factor of A and B, denoted GCF(A, B), is the largest number that is a factor of both A and B. For example,

GCF(2, 4) = 2
GCF(4, 2) = 2
GCF(60, 45) = 15
GCF(50, 45) = 5

There is an algorithm credited to Euclid, the ancient Greek mathematician, for calculating the GCF of two numbers. Repeatedly invoke the following formula until it yields a final answer:

$$\text{GCF(A, B)} = \begin{cases} A, if\ B = 0 \\ GCF(B, A\ \%\ B), otherwise \end{cases}$$

Rather than trying to explain why this algorithm works, it will be more instructive for you to code it up and run it with several examples to see it in action. Please write a program called **Euclid.java** that implements the above recursive algorithm. The program shall input the numbers A and B as two separate command-line arguments (`args[0]` and `args[1]`), and it shall print out the GCF of A and B. Use our usual error handling rules: if there is a problem with the user input, print "ERROR" to `System.err`, along with an explanation of the error.

## Binary Tree

The **Lab22** directory in the CourseInfo repository contains a Java file called **TestTree.java** that builds a binary tree and fills it with random values. Your job is to finish the code for two methods, both of which should be recursive.

- One method is used to add up all of the integer values stored in the tree
- The other method is used to find the minimum (most negative) value in the tree

## Reminder

Put all your files in your **Lab22** directory and push to GitHub before the deadline. This assignment must be turned in before 11:59pm Friday Nov 18th.

## Conclusion

The experience from this week's lab with recursion will help with the course's second large project.

## Grading Rubric

**Euclid.java** is worth 10 points:  2 points for each of 5 test cases

**TestTree.java** is worth 15 points:  5 points if it compiles, runs, and gives proper output.  0-10 points for code quality, including software design quality, code clarity, comments, and correctness of code.